

**SYSTEM AND METHOD FOR ANALYZING STATISTICS IN A REPORTING SYSTEM**

**Field of the Invention**

5           This invention relates to monitoring and analyzing statistics regarding informational and transactional content from a system, such as an reporting system, where the informational and transaction content may be provided to individuals through use of an output delivery system to email, web page, pager, mobile phone, fax, telephone, personal digital assistants and other terminal devices.

10           **Background of the Invention**

Decision support systems have been developed to efficiently retrieve selected information from data warehouses. One type of decision support system is known as an on-line analytical processing system. In general, business intelligence, OLAP or reporting systems analyze data from various perspectives and support complex analyses against large input data sets.

15           Reports may be generated by obtaining information and providing the report to a user. A report may be provided in a client-server relationship, over a network access (*e.g.*, posted on a web-site), or delivered directly to the subscriber. Current report systems, however, do not adequately track the statistics of the report or the informational and transactional content  
20           included therein.

          These and other drawbacks exist.

### **Summary of the Invention**

The invention relates to a system and method for capturing and analyzing statistics or data related to operation of a report system. Informational and transactional content is provided to a subscriber (also referred to a user), such as via a web posting, e-mail, voice and other channels.

5 Content may include informational, message or other forms of content, and may be in the form of a report. The captured statistics or data enables analysis of the report execution, as well as analysis of the report system itself.

According to an embodiment of the invention, content is aggregated from multiple data sources and services, such as databases, web pages, ERP Systems, XML based data sources and other sources. The aggregated content may be presented on a single page or report in an open source manner to a subscriber.

Statistics or data regarding the report, including the informational and transactional content included therein, and statistics or data regarding the report system itself may be captured and analyzed. Using analysis software or techniques, the statistics or data may be analyzed.

15 Analysis may include determining system performance, application usage and optimal configurations for the system. Statistics or data may be obtained in real-time, thereby enabling an end user to view the report system in real-time.

As an example of the present system, a report may provide information about investments. Informational and transactional content related to the investments may be  
20 aggregated into a content report, such as stock prices, interest rates, and a portfolio value. The report system provides the content report to the subscriber, such as through use of an output delivery system to email, web page, pager, mobile phone, fax, telephone, personal digital

assistants, WAP devices and other terminal devices. Also, a user may be able to access the report via a client-server or otherwise access the report from a reporting system, such as an OLAP system. Statistics or data about the report and the content included therein are captured and analyzed, enabling the profiling and understanding of current and historical system activity.

5 Other objects and advantages exist for the present invention.

### **Brief Description of the Drawings**

Fig. 1 is a block diagram illustrating an architecture for a reporting system according to an embodiment of the invention.

Fig. 2 is a flowchart illustrating steps performed by a process utilizing a query engine according to an embodiment of the invention.

Fig. 3 is a block diagram illustrating an architecture for a statistics analysis system according to an embodiment of the invention.

### **Detailed Description of the Present Invention**

The present invention provides for capturing statistics or data (hereinafter collectively referred to as statistics) to provide a view of server activity, such as, but not limited to, user sessions, reports, objects on reports, documents, and caches at different levels. Logged statistics may be analyzed, where one or more reports may be created based on the analysis. Reports based on this data module may be provided to an end user for overall enterprise monitoring. Capturing relevant statistics from a reporting system may be used to profile and understand current system activity, as well as historical system activity. Relevant statistics and data may include, but are

not limited to, overall system usage, individual user activity, usage data about reports, usage data about one or more objects used to operate and interact with the reporting system, database usage data (*e.g.*, workload, response times, the number of connections to the database, *etc.*), and concurrency information (*e.g.*, the number of concurrent users, the number of concurrent requests, the number of requests processed concurrently, the number of requests in the queue at any given time, *etc.*). Relevant statistics and data may also include server statistics (*e.g.*, amount of memory, computer processing unit (CPU) and input/output resources utilized at a given point) and metadata usage (*e.g.*, the number of requests for metadata, the types of requests for metadata, *etc.*). Other relevant statistics and data may also be used. For purposes of understanding the operation of the present invention, a description of a reporting system, such as an OLAP system, in which this invention may be utilized is provided below.

Fig. 1 is a block diagram illustrating a system 100 by which a variety of data resources may be accessed for business analytic, report generation and other intelligence purposes according to an embodiment of the invention. According to a preferred embodiment, the system 100 may comprise an Online Analytical Processing (OLAP) decision support system (DSS). In particular, Fig. 1 may comprise a portion of the MicroStrategy 7 or 7.1 platform, or future versions, which provides a preferred system in which the present invention may be implemented. Other platforms may also be used.

In general, through using the system 100 of the invention, analysts, managers and other users may query or interrogate a plurality of databases or database arrays to extract demographic, sales, and/or financial data and information and other patterns from records stored in such databases or database arrays to identify strategic trends. Those strategic trends may not be

discernable without processing the queries and treating the results of the data extraction according to the techniques performed by the systems and methods of the invention. This is in part because the size and complexity of some data portfolios stored in such databases or database arrays may mask those trends.

5 In addition, system 100 may enable the creation of reports or services that are processed according to a schedule. Users may then subscribe to the service, provide personalization criteria and have the information automatically delivered to the user, as described in U.S. Patent No. 6,154,766 to Yost *et al.*, which is commonly assigned and hereby incorporated by reference.

As illustrated in Fig. 1, a business, a government or another user may access the resources of the system 100 using a user engine 102. The user engine 102 may include a query input module 116 to accept a plurality of searches, queries or other requests, via a query box on a graphical user interface (GUI) or another similar interface. The user engine 102 may communicate with an analytical engine 104. The analytical engine 104 may include a set of extensible modules to run a plurality of statistical analyses, to apply filtering criteria, to perform a neural net technique or another technique to condition and treat data extracted from data resources hosted in the system 100, according to a query received from the user engine 102.

The analytical engine 104 may communicate with a query engine 106, which in turn interfaces to one or more data storage devices 108a, 108b ... 108n (where n is an arbitrary number). The data storage devices 108a, 108b ... 108n may include or interface to a relational database or another structured database stored on a hard disk, an optical disk, a solid state device or another similar storage media. When implemented as databases, the data storage devices 108a, 108b ... 108n may include or interface to, for example, an Oracle™ relational database

such as sold commercially by Oracle Corporation, an Informix™ database, a Database 2 (DB2™) database, a Sybase™ database, or another data storage device or query format, platform or resource such as an OLAP format, a Standard Query Language (SQL) format, a storage area network (SAN), or a Microsoft Access™ database. It should be understood that while data storage devices 108a, 108b ... 108n are illustrated as a plurality of data storage devices, in some embodiments the data storage devices may be contained within a single database or another single resource.

Any of the user engine 102, the analytical engine 104 and the query engine 106 or other resources of the system 100 may include or interface to or be supported by computing resources, such as one or more associated servers. When a server is employed for support, the server may include, for instance, a workstation running a Microsoft Windows™ NT operating system, a Windows™ 2000 operating system, a Unix operating system, a Linux operating system, a Xenix operating system, an IBM AIX™ operating system, a Hewlett-Packard UX™ operating system, a Novell Netware™ operating system, a Sun Microsystems Solaris™ operating system, an OS/2™ operating system, a BeOS™ operating system, a MacIntosh operating system, an Apache platform, an OpenStep™ operating system, or another similar operating system or platform. According to one embodiment of the present invention, analytical engine 104 and query engine 106 may comprise elements of an intelligence server 103.

The data storage devices 108a, 108b ... 108n may be supported by a server or another resource and may, in some embodiments, include redundancy, such as a redundant array of independent disks (RAID), for data protection. The storage capacity of any one or more of the data storage devices 108a, 108b ... 108n may be of various sizes, from relatively small data sets

to very large database (VLDB)-scale data sets, such as warehouses holding terabytes of data or more. The fields and types of data stored within the data storage devices 108a, 108b ... 108n may also be diverse, and may include, for instance, financial, personal, news, marketing, technical, addressing, governmental, military, medical or other categories of data or information.

5           The query engine 106 may mediate one or more queries or information requests from those received from the user at the user engine 102 to parse, filter, format and otherwise process such queries to be submitted against the data contained in the data storage devices 108a, 108b ... 108n. Thus, a user at the user engine 102 may submit a query requesting information in SQL format, or have the query translated to SQL format. The submitted query is then transmitted via  
10 the analytical engine 104 to the query engine 106. The query engine 106 may determine, for instance, whether the transmitted query may be processed by one or more resources of the data storage devices 108a, 108b ... 108n in its original format. If so, the query engine 106 may directly transmit the query to one or more of the resources of the data storage devices 108a, 108b ... 108n for processing.

15           If the transmitted query cannot be processed in its original format, the query engine 106 may perform a translation of the query from an original syntax to a syntax compatible with one or more of the data storage devices 108a, 108b ... 108n by invoking a syntax module 118 to conform the syntax of the query to standard SQL, DB2, Informix™, Sybase™ formats or to other data structures, syntax or logic. The query engine 106 may likewise parse the transmitted query  
20 to determine whether it includes any invalid formatting or to trap other errors included in the transmitted query, such as a request for sales data for a future year or other similar types of

errors. Upon detecting an invalid or an unsupported query, the query engine 106 may pass an error message back to the user engine 102 to await further user input.

When a valid query such as a search request is received and conformed to a proper format, the query engine 106 may pass the query to one or more of the data storage devices 108a, 108n ... 108n for processing. In some embodiments, the query may be processed for one or more hits against one or more databases in the data storage devices 108a, 108b ... 108n. For example, a manager of a restaurant chain, a retail vendor or another similar user may submit a query to view gross sales made by the restaurant chain or retail vendor in the State of New York for the year 1999. The data storage devices 108a, 108b ... 108n may be searched for one or more fields corresponding to the query to generate a set of results 114.

Although illustrated in connection with each data storage device 108 in Fig. 1, the results 114 may be generated from querying any one or more of the databases of the data storage devices 108a, 108b ... 108n, depending on which of the data resources produce hits from processing the search query. In some embodiments of the system 100 of the invention, the results 114 may be maintained on one or more of the data storage devices 108a, 108b ... 108n to permit one or more refinements, iterated queries, joinders or other operations to be performed on the data included in the results 114 before passing the information included in the results 114 back to the analytical engine 104 and other elements of the system 100.

When any such refinements or other operations are concluded, the results 114 may be transmitted to the analytical engine 104 via the query engine 106. The analytical engine 104 may then perform statistical, logical or other operations on the results 114 for presentation to the user. For instance, the user may submit a query asking which of its retail stores in the State of New



York reached \$1M in sales at the earliest time in the year 1999. Or, the user may submit a query asking for an average, a mean and a standard deviation of an account balance on a portfolio of credit or other accounts.

The analytical engine 104 may process such queries to generate a quantitative report 110, which may include a table or other output indicating the results 114 extracted from the data storage devices 108a, 108b ... 108n. The report 110 may be presented to the user via the user engine 102, and, in some embodiments, may be temporarily or permanently stored on the user engine 102, a client machine or elsewhere, or printed or otherwise output. In some embodiments of the system 100 of the invention, the report 110 or other output may be transmitted to a transmission facility 112, for transmission to a set of personnel via an email, an instant message, a text-to-voice message, a video or via another channel or medium. The transmission facility 112 may include or interface to, for example, a personalized broadcast platform or service such as the Narrowcaster™ platform or Telecaster™ service sold by MicroStrategy Incorporated or another similar communications channel or medium. Similarly, in some embodiments of the invention, more than one user engine 102 or other client resource may permit multiple users to view the report 110, such as, for instance, via a corporate intranet or over the Internet using a Web browser. Various authorization and access protocols may be employed for security purposes to vary the access permitted users to such report 110 in such embodiments.

Additionally, as described in the '766 Patent, an administrative level user may create a report as part of a service. Subscribers/users may then receive access to reports through various types of data delivery devices including telephones, pagers, PDAs, WAP protocol devices, email, facsimile, and many others. In addition, subscribers may specify trigger conditions so that the

subscriber receives a report only when that condition has been satisfied, as described in detail in the '766 Patent. The platform of Fig. 1 may have many other uses, as described in detail with respect to the MicroStrategy 7 and 7.1 platform, the details of which will be appreciated by one of ordinary skill in the reporting and decision support system art.

5           The steps performed in a method 200 for processing data according to the invention are illustrated in the flowchart of Fig. 2. In step 202, the method 200 begins. In step 204, the user may supply input, such as a query or a request for information, via the user engine 102. In step 206, the user input query may be preliminarily processed, for instance, to determine whether it includes valid fields and for other formatting and error-flagging issues. In step 208, any error conditions may be trapped and an error message presented to the user, for correction of the error conditions. In step 210, if a query is in a valid format, the query may then be transmitted to the analytical engine 104.

10           In step 212, the analytical engine 104 may further process the input query as appropriate to ensure the intended results 114 may be generated to apply the desired analytics. In step 214, the query engine 106 may further filter, format and otherwise process the input query to ensure that the query is in a syntax compatible with the syntax of the data storage devices 108a, 108b ... 108n. In step 216, one or more appropriate databases or other resources within the data storage devices 108a, 108b ... 108n may be identified to be accessed for the given query.

15           In step 218, the query may be transmitted to the data storage devices 108a, 108b ... 108n and the query may be processed for hits or other results 114 against the content of the data storage devices 108a, 108b ... 108n. In step 220, the results 114 of the query may be refined, and intermediate or other corresponding results 114 may be stored in the data storage devices 108a,

108b ... 108n. In step 222, the final results 114 of the processing of the query against the data storage devices 108a, 108b ... 108n may be transmitted to the analytical engine 104 via the query engine 106. In step 224, a plurality of analytical measures, filters, thresholds, statistical or other treatments may be run on the results 114. In step 226, a report 110 may be generated. The report 110, or other output of the analytic or other processing steps, may be presented to the user via the user engine 102. In step 228, the method 200 ends.

The present invention provides for capturing relevant statistics and data to enable profiling and/or understanding of current and historical activity of the reporting system.

Statistics may be stored, where storage methods disk based, file system based, database based, object oriented database based (*e.g.*, relational databases) or others, such as, but not limited to, NT Performance Monitor™ and NT Events Log™. Stored statistics may enable additional analysis of a reporting system activity, including identifying trends. Analyzing historical report system activity may include the ability to utilize decision support techniques and/or software to facilitate analysis.

Using and analyzing statistics from a report system may enable the report system to be optimized and tuned, such as various parts of the reporting system, including interfaces operating systems, databases, and informational sources. The present invention may enable identification of resource bottlenecks, spikes in user activities, cache hit ratios, the effectiveness of existing caches, recommendations for new caches and other changes to the reporting system. Statistics may also facilitate “charge-backs” and enable allocation of costs of the reporting system to various subscribers (within and outside of an organization) based on system usage.

According to an embodiment of the invention, a server used in connection with system

100 of Fig. 1 may record statistics, such as administrative statistics. Administrative statistics may be available for system administrators to analyze the operation of a server. Further, the server may provide for management of statistics, such as the ability to turn statistics logging on and off. According to an embodiment of the invention, one device for analyzing statistics is  
5 through a monitor which enables a user to view statistics gathered from the report system. A monitor may include one or more predetermined requests, where an end user may request particular statistics for evaluating the report system. Predetermined requests may include, but are not limited to, the top ten reports most provided to subscribers, web access trends, peak time periods, and job time-out reviews. In particular, a monitor such as MicroStrategy's Enterprise Manager™ may be used. To provide statistics monitoring, the present invention may provide a statistics module may collect data for a session module, a report module, a report detail module, a cache module and a document module. Data may be collected for other modules as well.

According to an embodiment of the invention, statistics may be gathered in real-time, thereby enabling an end user to analyze the reporting system in real-time or near real-time.  
15 Statistics reports may be generated in real-time or near real-time and then updated as new statistics are received, thereby providing a snapshot of the reporting system.

A system administrator may use statistics to monitor, configure, and tune a system. Statistics may enable analyses of system performance, indicate application usage and allow optimization of configurations within the system, as well as answer various questions regarding  
20 these topics. Statistics may be used for other analysis as well.

Analysis of system performance may include monitoring the status of the server.  
According to an embodiment of the invention, one or more transmission applications, such as a

Narrowcaster® or Telecaster® applications by MicroStrategy, may be used in conjunction with a monitor so that a system administrator may be alerted when certain conditions occur. According to an embodiment of the invention, third party systems management software may be used in connection with the present invention, but is not limited to, Computer Associates Unicenter™, Microsoft's Manager Console™, NetIQ™, and Tivoli™. Analysis of system performance may also include linking with other performance metrics (*e.g.*, via a performance monitor), understand caching effectiveness, understanding datamart effectiveness, understanding scheduling effectiveness, and tuning structural query language (SQL). According to an embodiment of the invention, statistics may enable an administrator to answer questions regarding system performance, including identifying system bottlenecks, whether an additional server should be added, whether server functions may be spread over multiple machines, how a metadata database server is impacting the system, and whether the metadata database should be moved to a separate machine. According to an embodiment of the invention, automated tuning of the report system may occur based on statistics and statistics reports. Automated tuning may include, but is not limited to, adding or removing resources from an analytical engine or a query engine. Other analysis and questions may also be performed and determined.

Indicating application usage may include indicating report and/or document usage, user activity, such as drill-downs and prompts, user access patterns (*e.g.*, time of day, day of month, *etc.*), billing functions, such as system resources used by a user and other usage. According to an embodiment of the invention, statistics may enable an administrator to answer questions regarding application usage, including what tables a query is hitting, what additional aggregate tables are needed, whether a larger percentage of reports should be cached, what reports/objects

are being used, what does the query access pattern look like, where does the database need tuning, and how could reports be designed more efficiently. Statistics may also include information on actions taken by a subscriber upon receiving a report from the reporting system. Actions may include, but are not limited to, deleting a report, accessing the report, forwarding the report and instructing a transaction based on the report. Other analysis and questions may also be performed and determined.

Optimization of a system's configuration may include determining optimal processing units structure, thread allocation structure, priority servicing scheme, prioritization functions, and governing parameters. According to an embodiment of the invention, statistics may enable an administrator to answer questions regarding optimal system configuration, including whether governing parameters should be changed, and whether the DSS Server should be tuned to run faster. Other analysis and questions may also be performed and determined.

According to an embodiment of the invention, an administrator may have a consistent view of all server-related statistics, regardless of the interface used to view them. That is, information stored in the statistics database may be consistent with similar measurements provided through the other sources of information. such as an NT Performance counter. By way of example, if a report regarding an engine mode in a reporting system client-server arrangement, indicates that it took 3 seconds to generate a SQL, then the same figure should be recorded in the statistics database. Other sources of information may include a server statistics database, including derived projects including, but not limited to a monitor, NT Performance Monitor, monitoring interfaces in the Server Administrator API, monitoring functionality in the Server Administrator GUI, and information presented in "Engine Mode" of the reporting system client-

server system. Other sources of information may also be available.

According to an embodiment of the invention, statistics collection may be supported on any platforms, such as platforms which support a metadata repository (*e.g.*, MicroStrategy's Narrowcaster® platform). Platforms may include MS SQL Server 7.0, Oracle 7.3, Oracle 8.0,  
5 Oracle 8I, and DB2 UDB 5.2. Statistics may be also supported for other platforms.

A server configuration application may create statistics tables. According to an embodiment of the invention, statistics tables may be created in a similar manner to the way repository tables are created. Separate DDL scripts may be available for each platform.

Statistics may be configured at a server-project level.

According to an embodiment of the invention, an administrator may prefer certain requirements. By way of example, different database servers may be used for the statistics database and the metadata repository. The database server may not necessarily be the same for both. By way of another example, an administrator may determine what database connections are being used for purposes of logging statistics. By way of another example, some customers may use a warehouse database server to host the statistics database. Other requirements may also  
10  
15  
be used.

The amount of statistics logging may be configurable. Statistics can be categorized into groups to enable configuration. Categorizing statistics may limit any performance impact of gathering statistics as well as the storage requirements. The categories used may depend on the  
20 design by an administrator, but may include user sessions, metadata object requests, element browse requests, report requests, and document requests. Other categories may also be used.

Administrators may selectively purge statistics (*i.e.*, delete records) from a statistics

database. The functions supported for purging statistics may include purging by date, purging by  
server and date, and purging by project and date. Purging by date may enable deletion of all  
statistics generated prior to a specific date or all statistics generated between two dates. Purge by  
server and date may enable deletion of statistics generated by a specific server. Purge by project  
5 and date may enable deletion of statistics generated for a specific project. Other functions for  
purging statistics may also be used.

According to an embodiment of the invention, in order to provide the real-time alert  
reporting to an administrator, statistics may be logged within a specified time period (*e.g.*, 5  
minutes) of collection time. Other time periods may also be used. A user may view statistics in  
10 real-time, where the statistics are generated and updated in real-time, as reports and contents are  
provided through the system being monitored. Statistics reports may be continuously written in  
real-time as statistics are gathered. This may enable statistics to be tracked as the content is  
provided (*e.g.*, as content streams from the server) without having to store statistics. According  
to another embodiment of the invention, statistics may be built up (*e.g.*, stored for some  
15 predetermined time) before being written to a statistics report.

Multiple servers may all use a single set of statistics tables, such as for the case of  
clustered configurations. Statistics tables may support analysis of all requests going through the  
cluster as a whole and analysis of requests going through a single node in the cluster. Other  
configurations of statistics tables may also be used.

20 According to an embodiment of the invention, for the statistics to be meaningful, the  
processing that a server performs may be defined for an end user of the statistics. For purposes of  
collecting statistics, various categories of information may be tracked, including user sessions,



metadata object requests, element browse requests, report execution requests, and document execution requests. Other categories may also be used.

A user may interact with a server in the context of a user session. Within a user session, the user may submit categories of requests to the server. For each type of request, the server may perform a sequence of tasks, which may be referred to as the executive flow. For some tasks, the server may queue the requests, *e.g.*, there may be queue time associated with some tasks. In a request for metadata object(s), a client may submit the request to a server. The server receives the request, creates a job (*e.g.*, the request for metadata objects) and starts it. The server processes the object request. Processing the object request may have a queue time associated with it. The server passes the result back to client, where the client receives the result.

According to an embodiment of the invention, results may be in the form of one or more statistics reports. Statistics reports may include, but are not limited to, activity by user, average execution time vs. number of jobs per user, average execution time vs. number of sessions, cache analysis, database result rows by user, database result rows by report, execution cycle breakdown, job cancelled review, job interruption trend, job performance trend, peak time periods, scheduled report load, server activity analysis, top ten longest executing reports, top ten reports, web access trends, web and non-web usage, and web usage statistics. By way of example, if cache analysis report is requested, column metrics for the report may include, but are not limited to, the number of jobs with cache creation, the number of jobs with a cache hit and the number of jobs without a cache hit, while row metrics may include, but are not limited to, all open issues by severity and by issue type overall and by team, all open issues by severity, and all open issues by issue type, severity and event, overall, by team and by page. By way of another example, if a database

results rows by report is requested, column metrics may include, but are not limited to, the number of jobs, the number of database result rows and the average elapsed duration per job, while row metrics may include, but are not limited to, issue list, issue count, all-technology issue count, issue count over time, count assigned or unfixed, general count, count RTT, count TBC, issue count by report, daily open issue counts, components, enterprise monitor issue counts, web report creation, enterprise monitor issues, and team counts. In an execution cycle breakdown request, column metrics may include, but are not limited to, average queue duration per job, average analytical engine duration, average SQL execution duration, and average SQL generation duration, while row metrics may comprise dates or other indications. If an average execution time vs. the number of jobs per user is requested, the column metric may include, but is not limited to, average number of jobs per user and average elapsed duration per job, while the row metric may comprise dates or other indications.

A cache request may be used to analyze statistics related to caching within the reporting system. Caching may occur within various components of the reporting system, such as, but not limited to, a web server, an application server, and a database server. By looking at the caches, such as by viewing the top ten cache hits, a report system may be more thoroughly analyzed. Analysis may include, but is not limited to, determining if caching is efficiently used within the report system.

Fig. 3 illustrates a system 300 for capturing statistics and data from a reporting system for analyzing and understanding current and historical reporting system activity. Statistics processor module 302 interacts with report system 100 from Fig. 1 to gather statistics. Statistic report generator module 304 generates statistics reports. User module 314 of system 300 may provide a

request for one or more statistics through input/output module 310. User module 314 may be a computer processor, PDA, or any other device which enables a person to interact with system 300. A request may be stored in queue 312 in input/output module 310. Queue 312 may also be located in other modules within system 300.

5           Based on a request received at input/output module 310, statistics report generator module 304 may generate a statistics report using gathered statistics. According to an embodiment of the invention, statistics may accumulate in cache module 308 before a statistics report is generated by statistics report generating module 304. Statistics gathered by statistics processor module 302 and statistics reports generated by statistics report generating module 304  
10           may be stored in storage module 306. Multiple storage modules 306 may be used to store statistics and statistics reports. Statistics reports may be transmitted by input/output module 310 to an end user module 316. End user module 316 may be a computer processor, PDA, or any other device which enables a person to interact with system 300 to view a statistics report. Other configurations for system 300 may also be used, as well as other system for capturing and  
15           analyzing statistics.

          An element browse request may enable an end user to various elements provided by the system. By way of example, a client submits the request, which is received by the server. The server creates a job and starts it. Information for the job is resolved, the server generates a SQL and executes the job (which may include SQL execution and analytical processing). The element  
20           results are prepared by the server. The server passes the result back to client, which receives the results. One or more steps with this process may have a queue time associated with it. According to an embodiment of the invention, if an element browse request uses the cache, the

request may go through a slightly different execution flow. A client submits a request to a server. The server creates a job and starts it. The server prepares the element results, and passes the result back to client. One or more of these steps may have a queue time associated with it.

A report execution request may reflect an end user's view of the execution flow for the  
5 executing of one or more report(s). A client submits a request to a server. The server creates a job, starts it and resolves information for the job. The server generates a SQL, and executes the job (which may include SQL execution and analytical processing). The server crosstabs or reformats the results, and passes result back to client. If a cached report is executed, the client may submit a request to a server. The server passes results back to the client. One or more of  
10 these steps may have a queue time associated with it.

If a report request contains prompts, the request may be submitted to the server by the client. The server creates a job, starts it, and resolves information for the prompt. The server then passes the prompt back to client, where the client receives a prompt question. The client then submits a second request, which then follows the execution flow of a "normal case,"  
15 described above. Thus, a request for a report with prompts may result in two statistics records: one for the resolving the prompt and one for executing the report once the answers are given. One or more of these steps may have a queue time associated with it.

A scheduled report may proceed through the same execution flow as a report request submitted by a remote client. For example, such an execution flow may include a scheduler  
20 submitting a request to a server. The server creates a job, starts it and resolves information for the job. The server generates a SQL and executes the job (which may include SQL execution and analytical processing). The server may cross-tab the results and pass the results back to the

client or scheduler, who receives result. One or more of these steps may have a queue time associated with it.

According to an embodiment of the invention, the execution flow for a document execution request may include a client submitting a request to the server. The server may create  
5 a job, start it, and resolve report information for the job. The server executes the document (which may include multiple report executions) and pass the results back to the client. The server may spawn additional jobs/requests for a document's constituent reports. Each of these additional jobs/requests may follow the execution flow described above for report requests. Therefore, a single document request may result in  $N+1$  records in the statistics database, where  
10 one record is for the document request itself and  $N$  is for the constituent reports, where  $N$  is the number of reports. One or more of these steps may have a queue time associated with it. Other execution flows for other requests may be used.

The specifications for statistics database tables may be maintained in a separate document. According to an embodiment of the invention, this document may specify the exact  
15 information required for each type of request described above. In addition, the document may define all fields in all tables, their meanings, and the corresponding DDL for all supported platforms. Other specifications may also used.

Other embodiments and uses of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The  
20 specification and examples should be considered exemplary only.